NO-94-23-4 ASHRAE Transactions, Vol. 100, No. 1, 1994

USING EMULATORS TO EVALUATE THE PERFORMANCE OF BUILDING ENERGY MANAGEMENT SYSTEMS

G.E. Kelly, Ph.D.
Member ASHRAE

W.B. May, P.E.

J.Y. Kao, P.E.
Member ASHRAE

C. Park, Ph.D.

ABSTRACT

The performance of a building energy management system (BEMS) is directly related to the amount of energy consumed in a building and the comfort of the building's occupants. One approach with which to evaluate the performance of a BEMS is through the use of an emulator—a special computer/data acquisition system that is connected to the sensor inputs and command outputs of the BEMS. It replaces the building and its heating, ventilating, and air-conditioning (HVAC) systems and uses a computer program to simulate their response to BEMS commands. The BEMS, through its supervisory and/or direct digital control algorithms, then controls the simulated building/HVAC system as if it were an actual one. At the same time, the emulator evaluates the performance of the BEMS in terms of the energy consumed by the simulated building, the degree of comfort maintained in the simulated space, response time, accuracy, etc.

This paper describes using emulators to evaluate a BEMS. Major topics include setting up a BEMS and an emulator, evaluating system/command and DDC software, and methodologies for testing BEMS application algorithms. Considerations are presented for evaluating the programming capabilities of a BEMS, DDC control loop performance, and rating different aspects of BEMS performance. A brief discussion of BEMS software is also included.

INTRODUCTION

An emulator for building energy management system (BEMS) applications consists of a computer-based simulation of a building and its mechanical system connected to a real BEMS (May and Park 1985). It can be used to replace the entire building/HVAC/plant system, or the emulation software can be interfaced with selected pieces of real HVAC hardware, such as coils, valves, a boiler, or a chiller. The emulator can then be used for evaluating the performance of a BEMS before or after its purchase (Vaezi-Nejad et al. 1991). By varying the tests performed, an emulator can be used to evaluate either a BEMS that will be installed in a particular building or a BEMS that a large organization may consider purchasing for installation in many different types of buildings in various climates.

An emulator is connected to the BEMS being evaluated in place of the regular BEMS sensors and actuators. Since most BEMS sensors are electrical, they can be replaced with voltage and current sources under control of the emulator. The BEMS is unable to distinguish between an actual sensor producing an electrical signal and the emulator producing the same electrical signal. The BEMS, through its supervisory and/or direct digital control (DDC) algorithms, then controls the emulator building/HVAC system as if it were an actual one. At the same time, the emulator evaluates the performance of the BEMS in terms of the energy consumed by the simulated building, the degree of comfort maintained in the simulated space, the response time, accuracy of control, etc. This allows the BEMS software algorithms to be evaluated directly without the effect of sensors distorting the results. Figure 1 shows the arrangement of an emulator and a BEMS.

An advantage of using an emulator is that a BEMS may be tested with any type of building/HVAC/plant system for which a simulation model is available, and tests can be repeated on different BEMS under identical conditions. In addition, it is not necessary to know the exact structure of the algorithms used in a BEMS to evaluate how well they perform. Since BEMS software is usually proprietary, this is a definite advantage.

BUILDING ENERGY MANAGEMENT SYSTEM SOFTWARE

In general, there are four basic types of software associated with BEMS that an emulator can be used to evaluate. System software consists of the operating systems and utility programs of the BEMS host computer, the field panels, and unitary controllers. Command software includes explanations of commands, an index of commands, commands to define and modify physical parameters and constraints assigned to monitoring and control points, commands to request reports and graphic displays, identifi-

George E. Kelly is a group leader, and **James Y. Kao** and **Cheol Park** are mechanical engineers at the National Institute of Science and Technology, Gaithersburg, MD. **William B. May**, who worked for NIST while this work was being performed, is currently vice president of Technical Services, MEDMIS, Arlington, VA.

TABLE 1
Typical Output and Input Points of an Emulator for an Air-Handling System

Output Points from Emulator Tester	Input Points to Emulator
1. Zone space dry-bulb temperature	1. Start/stop AHU command
2. AHU status	2. Economizer on/off command
3. Zone space humidity (or equiv.)	3. Close/open min. vent. dampers
4. Outside air temperature	4. Purge on/off command
5. Building electrical demand	5. Local setback/normal command
6. Outside air humidity (or equiv.)	6. Setpoint resetting command
7. Return air dry-bulb temperature	
8. Return air humidity (or equiv.)	
9. Zone heating/cooling demand	

Test Condition Selection

The test conditions are those parameters that describe the simulated building and the environmental conditions to which it is exposed in order to emulate a real building system connected to the BEMS system. There are three basic types of test conditions that must be specified for the emulator: weather, building system type, and building use.

Weather is the most difficult of the test conditions to select. The weather experienced by a building will vary with location and time of year and generally is not predictable except on a long-term statistical basis. There are two parts to the selection of weather conditions. The first part is associated with the location of the building and describes the general range of weather conditions or the climate to which a building will be subjected within a year. If the BEMS is installed in a specific building at a given location, the choice of climate to use in the performance evaluation is already determined. If the BEMS is to be installed in buildings in a variety of geographical locations with vastly different climates, then performance tests may have to be conducted over the range of possible climates.

The second part of weather condition selection is the characterization of annual performance of a BEMS algorithm without having to test the algorithm for a year. For the most complicated algorithms, it should be sufficient to test the algorithm for one or more days in each season of the year. What is needed is a weather description (based on either simulated or real data) for periods of high, moderate, and low loads. Such periods might correspond to the peak of summer and winter (July and January), early summer and early winter periods (May and October), and spring (April).

The variables to be included in a weather condition include dry-bulb temperature, humidity, solar gain, wind speed, and barometric pressure. Within a single day, the weather variables will change with time. There must be a specification of the way in which the variables change. A reasonably good approximation can be made using an outside air temperature that varies in a sinusoidal fashion. Another approximation is to use a constant humidity ratio for any particular day but use different humidity ratios in different seasons. If five different weather conditions were used to approximate a range of yearly conditions, this would imply at least five days of testing. If the building emulated were of high mass, the transition between days at different conditions might require several days to allow the temperature of the building structure to reach a steady oscillation. This could be avoided by reinitializing the structure temperatures as the weather made a transition into the new season. If optimum start/stop were being tested, however, several days might be needed for an adaptive-type algorithm to operate properly. For other algorithms, this would not be necessary. If weather conditions needed to be run for one day, and five conditions were needed, then five days of testing would be required.

The building system type will characterize its response to environmental conditions. The conditions can be divided into those describing the building shell and those conditions describing the HVAC system type. The conditions for these two types can be decoupled to some extent, since different HVAC systems could be installed in the same building shell.

The building shell characteristics include resistance to thermal heat transfer, resistance to infiltration, thermal capacity (mass), solar heat gain areas/shading factors/transapplication software. This type of testing should generally fall into two categories: operational testing and performance testing. In discussing these two test categories, it will be assumed that the application algorithms being evaluated are supplied as "canned" software by the BEMS manufacturer. If this is not the case and the application algorithms are programmed by the installer, the BEMS purchaser, or the evaluation team, the same performance tests may still be used but the evaluation team must keep in mind the fact that the results will be a measure of both the BEMS performance and the programming skill of the installer/purchaser/evaluation team. In such an instance, the suggested rating procedure described later in this paper may also need to be modified to provide a fair comparison between different BEMS.

Operational Testing

Operational testing of BEMS application algorithms consists mainly of exercising the different logical branches in each algorithm and determining if they operate as expected. Like the evaluation of operating system, command, and DDC software, operational testing of supervisory control algorithms, in most cases, is likely to involve tests of limited duration. However, tests are likely to include a variety of test conditions linked together to exercise the logic built into the application software. Although operational tests should usually be conducted on each application algorithm individually, they may also be performed on a combination of algorithms to determine if they interact in an acceptable manner. The BEMS evaluation team should have no difficulty selecting sequences of test conditions for an emulator that would operationally test most of the "typical" application algorithms found on today's BEMS. More advanced algorithms, which optimize overall building system performance, might best be evaluated using the performance testing methods discussed here.

Performance Testing

An emulator is essential for evaluating application programs that involve complex control over long periods of time or where there are strong interactions between such programs. To illustrate this, several supervisory control algorithms have been selected and a scenario is developed in this section for how an emulator might be used to evaluate their performance. Suggestions for performance testing other application programs may be found in Kelly et al. (1992).

One of the most important steps in evaluating the performance of a BEMS is selecting the application algorithms of interest and then deciding which combination of these algorithms is to be tested. These algorithms must be carefully selected and will be strongly dependent on the particular building/HVAC system being emulated. For example, the evaluation team might decide that the emulator should be used to evaluate the performance of the following six BEMS application programs involving control of a building's air-handling system: scheduled start/stop, duty cycling, demand limiting, optimum start/stop, economizer cycle, and ventilation/recirculation. To shorten the testing process and produce reasonably realistic tests for algorithms that must normally interact with each other, the testing procedures should test some algorithms concurrently. It will be assumed that the scheduled and optimum start/stop algorithm should turn an air-handling unit on during occupied periods and off during unoccupied periods. During occupied periods only, the duty-cycling algorithm should turn an air handler off and on. Also during occupied periods, the demand-limiting algorithm should cut off the air handler for short periods. If these strategies are not coordinated or prioritized, their control of the same load can cause problems. These problems, if they exist for a specific system under test, are likely to appear during the concurrent tests. The economizer and ventilation/recirculation algorithms both control the outside air damper operation and could be logically grouped together for concurrent testing. Thus, the tests to be conducted might be reduced to the following combination of algorithms: scheduled start/stop, duty cycling, and demand limiting; optimum start/stop, duty cycling, and demand limiting; and economizer cycle and ventilation/recirculation.

It is likely that the demand-limiting algorithm will have to be tested in a partial manner, since the true electrical demand of the building may not be known. In this case, a "simulated demand meter" might be connected to the BEMS for the purpose of simulating a demand level sufficient to cause the demand-limiting algorithm to shut down the air-handling unit.

Concurrent Testing

An emulator may also be used to perform different tests at the same time. For such concurrent testing, an important consideration is how many independent concurrent tests to run at the same time. If the emulator had sufficient computational power, more than one building could be emulated at the same time. The BEMS could then be configured to run the strategies on different physical buildings. The disadvantage of this approach is that more physical points must be connected to the emulator. The advantage is that the BEMS is loaded more completely, and more tests can be completed in less time. The number of concurrent tests possible will depend on the number of points available to the emulator and on the model used to emulate the building. More than one test will require multiple copies of the model programs and will require more memory and more computer execution time. The number of concurrent tests will depend on the power of the computer on which the emulator is based.

a two-day test period for each climatic condition, then the tests would be run for five two-day periods or a total of 10 days if five different climatic conditions were studied.

In case of failure during the test, the emulator software must be stopped. If the failure is in the BEMS algorithm, the problem must be corrected before proceeding. If the failure is in the emulator, the problem may require attention to the emulator hardware or, in the simplest case, the emulator may be restarted or reset to correct the problem. When the total test is divided into test segments, the test can be restarted with the segment that was under way at the time of the failure. After the time duration of a test segment has elapsed, the emulator should automatically store the intermediate results. If a forced stop occurred before the completion of the subtest, the emulator software should have a provision to stop the test. If there are five subtests to be run with different climatic conditions, the emulator operator may select two options for running the five subtests. The options are to either have the emulator software start the subsequent subtests automatically or to require a manual start of the other subtests. The other climatic condition subtests must be started at the same emulated time of day as the original subtest.

After all subtests are successfully completed, the BEMS algorithms being tested by the emulator should be stopped or disabled. The emulator should then be used to determine the test ratings. After ratings have been correctly obtained, the emulator may be turned off and disconnected from the BEMS. If no further tests are planned, the BEMS field panel may be removed and the points in this panel deleted from the BEMS point data base. When all climate subtests have been completed, the test ratings should be obtained to determine the performance of the algorithms tested. After each subtest, the energy consumed by the cooling and heating systems, as predicted by the building simulation in the emulator, should be stored on the emulator's disc storage. Also stored should be information on occupant comfort levels and maintenance requirements. In addition, reports of all of the command events generated by the BEMS and the times at which they occurred should be stored on the disc. This information should be used to determine the algorithm rating.

Criteria for Evaluation

The main purpose for installing BEMS algorithms for control of a building is to reduce energy consumption to levels below those of a building without BEMS algorithms. Therefore, energy savings should be the primary criterion by which an algorithm is judged. However, use of an algorithm may result in a net energy savings and yet produce occupant discomfort, incorrect or inconvenient activation or deactivation of equipment, or increased maintenance costs. The four main criteria by which the BEMS algorithms may be judged are energy savings, occupant comfort, maintenance requirements, and BEMS algorithm errors. An additional criterion for evaluation of one particular algorithm, demand limiting, is the reduction of electrical demand. This algorithm must be evaluated in terms of monetary savings rather than energy savings. Due to the difficulty of simulating electrical demand and demand billing structures, as defined by the local electric utility, only partial testing of the demand-limiting algorithm may be possible.

The energy saved by an algorithm is determined by comparing the energy use of a building between two cases: with and without the BEMS algorithm in operation. In order to evaluate energy savings of algorithms, the emulator software must contain a model of a building and its HVAC system, weather conditions, and occupancy schedules to which the model is subjected. The energy used by the building can be determined from the model output and can be categorized as energy used by the fans in the air-handling unit, energy used to heat or cool the air passing through the air-handling unit, energy used by local space-heating or space-cooling equipment (reheat coils, heat pumps, fan coils, perimeter radiation, air conditioners), and total energy used by the building system. The energy amounts reported may or may not be in terms of fuel consumption based upon assumed or measured plant efficiencies. In the latter case, the energy reported may represent the energy added to the air and extracted from steam/hot water for heating or the energy extracted from the air by chilled water or refrigerant for cooling. The energy used by the building without the algorithm(s) could be previously determined and stored in the emulator, or the test could include a period of testing with the algorithm deactivated. This would involve more time but might produce better results. Public domain algorithms could also be tested for a specific set of building types, use, and climate, and energy savings could be published as a set of tables or graphs for use in specifying minimum standards for algorithms.

The specification of occupant comfort is not well defined in the HVAC industry yet is important in testing BEMS algorithms. Two possible comfort criteria could be used. One is a statistical type of measure that predicts the percentage of people occupying a space who should be uncomfortable under given conditions. The predicted percentage of dissatisfied (PPD) is widely used for this type of measurement. Another possibility is the use of a comfort zone approach. The comfort zone method would allow two states of comfort. Inside a range of dry-bulb temperatures and humidity ratios, the comfort would be acceptable. Outside this range, the comfort would be unacceptable. Other criteria for the comfort zone method would be a range of mean radiant temperatures and a maximum rate of change of temperature or humidity. Eventually, it may be desirable to include air quality or fresh airflow rates. Due to the simplicity of the comfort zone approach, the comfort zone criteria may be the most suitable approach when control of different building systems is tested with an emulator. During a test of an algorithm, the predicted values of the temperature and humidity from the emulation

TABLE 3
A Tabulated Format to Show Test Results

	Early winter		Mid winter		Spring		Early summer		Mid summer	
Electrical energy										
	Heat	Cool	Heat	Cool	Heat	Cool	Heat	Cool	Heat	Cool
Economizer energy										
Load energy										
Reheat/recool energy										
Total energy use										

to be run without endangering any expensive equipment or interfering with normal building operations.

Criteria for evaluating the programming capabilities of a BEMS include

- the ease of use of the programming language;
- the amount of "canned" software provided with the BEMS and the difficulty of employing it in user-developed programs;
- the quality of local-loop control achieved;
- the convenience of using hierarchical control loops;
- the capabilities of the editor, compiler, linker, and debugger routines provided with the BEMS;
- the ability to create user-developed graphics and to display data on the graphics in real time; and
- the ease of starting, stopping, editing, and changing parameters in user-developed programs.

DDC CONTROL LOOP PERFORMANCE

Most BEMS systems come with "canned" software for performing proportional (P), proportional plus integral (PI), and proportional plus integral plus derivative (PID) control. Some systems employ algorithms that will tune themselves upon command. A few may contain "adaptive" algorithms that continuously monitor and adjust to changes in load, actuator characteristics, etc. Most, however, require the BEMS operator to go through a tuning process to select the appropriate gains for a particular control loop.

For those algorithms that tune themselves or are adaptive in nature, the emulator should be used to evaluate their performance for a variety of control loops typically found in building applications. For example, different types of coils, dampers, and actuators could be emulated. Different values of process gain, dead time, hysteresis, and sensor mass could also be emulated, along with different degrees of nonlinearity. The time it takes to tune each loop and the quality of control achieved should be monitored to determine how well the algorithms perform.

For algorithms requiring manual tuning, the operator should actually perform the tuning process using the trial-and-error method, closed-loop tuning method, or one of the other methods described in Stoecker and Stoecker (1989). The emulator should be used to emulate a variety of control loops having the different characteristics described previously.

The criteria for evaluating the performance of different BEMS at local-loop control include control resolution obtained with different ranges of sensor input, control stability, control accuracy and repeatability, ease of use, flexibility of cascading control loops, and the hourly number of starts/stops/reversals of the actuator.

SUGGESTED BEMS RATING METHODOLOGY

Because each BEMS application is different and different evaluators will have different opinions, it is difficult to give a definitive rating scheme for comparing the performance of different BEMS. A possible approach is suggested here for a rating methodology. This suggested methodology assumes that the application algorithms are supplied as canned packages by the BEMS manufacturer. If this is not the case, the rating scheme should be altered to reflect the fact that the test results are an indication of both the performance of the BEMS and the skill of the persons (e.g., the installer, BEMS operator, or evaluation team) programming the application algorithms.

Rating factors should be determined first. They should include the appropriateness of both hardware and software, operational aspects (user friendliness, data sharing between field panels, etc.), alarm appropriateness, clarity and adequacy of reports, correct operation of command and

3. All the application algorithms operate correctly.
   Operational tests were performed on the following algorithms and their control logic was found to be correct: ________ ________ ________

   ____
   ____
   ____
   ____

   Problems were found with the operation of the following algorithms:

   ____
   ____
   ____
   ____

4. Performance testing of application algorithms indicate optimal energy savings and occupant comfort, with no major errors detected. ________ ________ ________
   Performance tests were performed on the following application algorithms and/or combination of application algorithms:

<u>Algorithm/Combination</u>	<u>Duration of Test</u>
__________	__________
__________	__________
__________	__________
__________	__________

   Were energy and comfort results compiled and compared for all of the above algorithms or combination of algorithms?
   Were interaction problems detected between or among different application algorithms?
   If so, list them:.

   ____
   ____
   ____
   ____

   The following errors were detected during the above performance tests:

   ____
   ____
   ____

5. The BEMS performance capabilities were evaluated and found to be acceptable. ________ ________ ________
   Evaluation factors considered included:
   - ease-of-use
   - quantity/quality of software "tools"
   - quality of control
   - hierarchal control loops
   - adequate editor, compiler, linker, and debugger
   - user can easily develop graphic displays and show data in real time
   - ease of starting, stopping, and editing user developed programs and making changes in assigned parameters